

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-155698

(43)Date of publication of application : 06.06.2000

(51)Int.Cl.

G06F 11/18
G06F 15/16

(21)Application number : 2000-014530

(71)Applicant : NEC COMMUN SYST LTD

(22)Date of filing : 28.02.1994

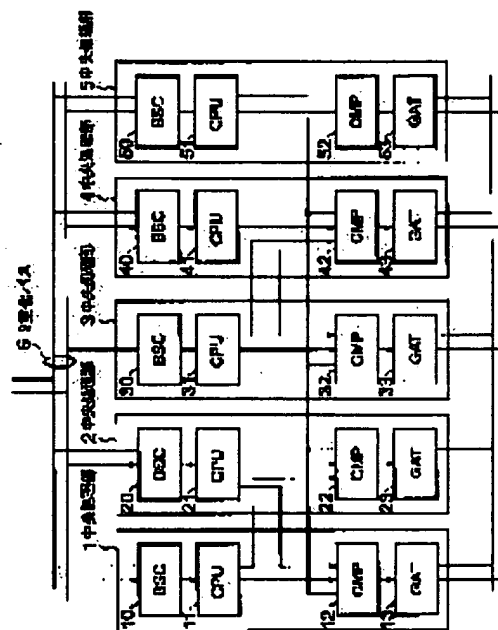
(72)Inventor : HASEGAWA NOBUYA

(54) FAULT TOLERANT SYSTEM OF COMPUTER

(57)Abstract:

PROBLEM TO BE SOLVED: To realize a fault tolerant system that operates a multitask processing computer with fault tolerant which simultaneously executes two or more task processing with a small number of central processing unit parts, operates without stoppage.

SOLUTION: Two current central processing parts 1 and 2 are used for one task processing, another spare central processing part 5 is further provided, the spare central processing part 5 executes the same task processing only when CMPs 12 and 22 compare task processing results made by the current central processing parts 1 and 2 and when they do not coincide, a GAT 13 or a GAT 14 is opened and, the processed results of a current CPU 11 or CPU 21 which coincide with the processed results made by a CPU 51 are outputted.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-155698

(P2000-155698A)

(43) 公開日 平成12年6月6日(2000.6.6)

(51) Int. Cl.	識別記号	F I	キーワード(参考)
G 0 6 F 11/18	3 1 0	G 0 6 F 11/18	3 1 0 C
			3 1 0 E
15/18	6 4 0	15/18	6 4 0 J

審査請求 有 請求項の数 2 O L (全 6 頁)

(21) 出願番号 特願2000-14530(P2000-14530)
 (62) 分割の表示 特願平4-29925の分割
 (22) 出願日 平成6年2月23日(1994.2.23)

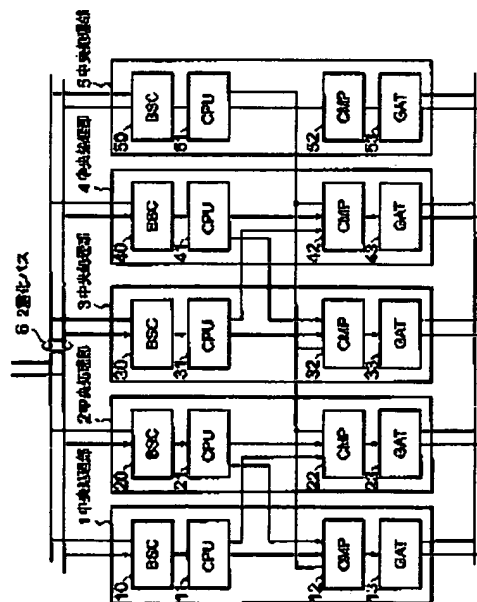
(71) 出願人 000232254
 日本電気通信システム株式会社
 東京都港区三田1丁目4番28号
 (72) 発明者 長谷川 伸弥
 東京都港区三田1丁目4番28号 日本電気
 通信システム株式会社内
 (74) 代理人 100082935
 弁理士 京本 直樹 (外2名)

(54) 【発明の名称】 コンピュータのフォールト・トレラント方式

(57) 【要約】

【課題】 小規模な中央処理装置数で、同時に二つ以上のタスク処理を実行するマルチタスク処理コンピュータを無停止で運転させるフォールト・トレラント方式を実現する。

【解決手段】 1つのタスク処理に対して2つの現用の中央処理部1、2を使用し、その他に1個の予備の中央処理部5を有し、現用の中央処理部1、2によるタスク処理結果をCMP12、22で比較して不一致となった場合のみ、予備の中央処理部5で同じタスク処理を実行して、CPU51による処理結果と一致した現用のCPU11またはCPU21の処理結果をGAT13またはGAT14を開いて出力する。



(2)

特開2000-155698

1

【特許請求の範囲】

【請求項1】同時に二つ以上のタスク処理を実行するマルチタスク処理コンピュータを障害の有無にかかわらず無停止運転させるフォールト・トレラント方式であつて、

第一及び第二の中央処理部から構成されて同時タスク処理を行うタスク数に対応した複数組の現用中央処理部と、一個の予備中央処理部とを備え、

前記各組の第一及び第二の中央処理部は一つのタスク処理を同期して並列に実行し、当該第一及び第二の中央処理部の出力結果が一致した場合のみ処理結果を出力し、第一及び第二の中央処理部の出力結果が不一致の場合は、前記予備中央処理部を起動して出力結果が不一致であった組の現用中央処理部が実行したタスク処理を実行し、当該予備中央処理部のタスク処理結果と一致する第一及び第二の中央処理部のいずれかの処理結果を出力することを特徴とするコンピュータのフォールト・トレラント方式。

【請求項2】同時に二つ以上のタスク処理を実行するマルチタスク処理コンピュータを障害の有無にかかわらず無停止運転させるフォールト・トレラント方式であつて、

同時タスク処理を行うタスク数に対応して設置され、各組が各タスク処理を同期して並列に実行する二つの中央処理装置と、当該中央処理装置のそれぞれに対応して設置されて相手側の中央処理装置と一致した処理結果が得られたときにのみ処理結果を出力する比較手段を備えた、複数組の現用中央処理部と、前記現用中央処理部のいずれかにおいて処理結果の不一致があった場合に起動され、処理結果の不一致が発生した現用中央処理部が実行したタスク処理を実行し、当該処理結果を処理結果の不一致が発生した現用中央処理部の前記比較手段に出力する予備の中央処理装置を有し、前記比較手段は、前記予備の中央処理装置の処理結果を受信し、対応する自中央処理装置の処理結果と一致するときに処理結果を出力することを特徴とするコンピュータのフォールト・トレラント方式。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】本発明はコンピュータのフォールト・トレラント方式に関する。

【0002】

【従来の技術】図3は従来のコンピュータのフォールト・トレラント方式の第1の例を示すシステムブロック図である。

【0003】図3を参照すると、この従来例のシステムでは、中央処理装置（CPU）101、102および103と比較回路（CMP）104とにより密結合な中央処理部100を構成している。

【0004】そして、1つのタスク処理をCPU10

2

1、102および103が同期して実行しており、それぞれCPU101、102および103の処理結果出力はCMP104によって多数決論理により比較され、最終的に一番多い処理結果を出力する。

【0005】この第1の従来例の方式では、例えばCPU101が障害となったとき、CPU101はCPU102、103とは異なる値を出力するので、CMP104はこの異なる値を比較して多数の方のCPU102およびCPU103の処理結果を優先して出力する。

【0006】したがって、CPU101の処理結果は無視される。このようにしてCPU101が障害の間もシステムの正常動作を維持している。

【0007】また、第2の従来例として、例えば特開平4-85028号公報に所載の「コンピュータのフォールト・トレラント方式」がある。図4は第2の従来例のシステムブロック図である。

【0008】図4を参照すると、このシステムは、バス417と接続されたゲート回路（GAT）418、419、420と、相互に同期して同一処理を行いGAT418、419、420にそれぞれ処理結果を出力する中央処理装置（CPU）411、412、413と、CPU411とCPU412との各出力、CPU412とCPU413との各出力、CPU413とCPU411との各出力をそれぞれ比較する第1、第2、第3の比較回路（CMP）414、415、416とを備えている。

【0009】この場合、各CPU411、412、413が正常であれば、CPU411、412、413の各出力が同一状態として、かつ、同期して出力バス417へ送出され、相互に干渉することなくデータ処理が行われる。

【0010】これに対し、例えば、CPU411に異常が生じると、CMP414とCMP416とが不一致出力を生じ、これにより、GAT418とGAT420とがオフするため、CPU411とCPU413との出力は阻止され正常なCPU412、413中の1台、即ち、CPU412の出力のみが出力バス417へ送出される。

【0011】

【発明が解決しようとする課題】この従来のコンピュータのフォールト・トレラント方式は、密結合な3つのCPUで1つの中央処理部を構成しているため、1つのタスク処理を実行するのに常に3つのCPUが必要となる。

【0012】このため、同時に2つ以上のタスク処理を実行するマルチタスク処理では、タスク数が増えるほどハードウェア構成が物理的に大規模になってしまうという問題点があった。

【0013】本発明の目的は、1つのタスク処理に対して2つの現用の中央処理部を使用し、その他に1個の予備の中央処理部を有し、2つの現用の中央処理部の処理

(3)

特開2000-155698

3

結果が不一致になったときは、予備の中央処理部を使用して再度処理を実行してタスク処理を引き継ぐことにより、フォールト・トレラント動作を可能にしたコンピュータのフォールト・トレラント方式を提供することにある。

【0014】

【課題を解決するための手段】本発明によれば、同時に二つ以上のタスク処理を実行するマルチタスク処理コンピュータを障害の有無にかかわらず無停止運転させるフォールト・トレラント方式であって、第一及び第二の中央処理部から構成されて同時タスク処理を行うタスク数に対応した複数組の現用中央処理部と、一個の予備中央処理部とを備え、各組の第一及び第二の中央処理部は一つのタスク処理を同期して並列に実行し、第一及び第二の中央処理部の出力結果が一致した場合のみ処理結果を出力し、第一及び第二の中央処理部の出力結果が不一致の場合は、前記の予備中央処理部を起動して出力結果が不一致であった組の現用中央処理部が実行したタスク処理を実行し、この予備中央処理部のタスク処理結果と一致する第一及び第二の中央処理部のいずれかの処理結果を出力することを特徴とするコンピュータのフォールト・トレラント方式が得られる。

【0015】また、同時に二つ以上のタスク処理を実行するマルチタスク処理コンピュータを障害の有無にかかわらず無停止運転させるフォールト・トレラント方式であって、同時タスク処理を行うタスク数に対応して設置され、各組が各タスク処理を同期して並列に実行する二つの中央処理装置と、当該中央処理装置のそれぞれに対応して設置されて相手側の中央処理装置と一致した処理結果が得られたときにのみ処理結果を出力する比較手段を備えた、複数組の現用中央処理部と、前記の現用中央処理部のいずれかにおいて処理結果の不一致があった場合に起動され、処理結果の不一致が発生した現用中央処理部が実行したタスク処理を実行し、当該処理結果を処理結果の不一致が発生した現用中央処理部の前記比較手段に出力する予備の中央処理装置を有し、前記の比較手段は、予備の中央処理装置の処理結果を受信し、対応する自中央処理装置の処理結果と一致するときに処理結果を出力することを特徴とするコンピュータのフォールト・トレラント方式が得られる。

【0016】

【発明の実施の形態】次に、本発明について図面を参照して説明する。

【0017】図1は本発明のコンピュータのフォールト・トレラント方式の第1の実施例を示すシステムブロック図で、シングルタスク処理を実行する例を示している。

【0018】図1を参照すると、本実施例のコンピュータシステムは、2重化バス6に接続された中央処理部1と中央処理部2とで構成された1組の現用の中央処理部

4

と1個の予備の中央処理部5とを備え、各中央処理部、例えば中央処理部1は2重化バス6とインタフェースするバス・コントロール部(BSC)10と、BSC10を通して送られたタスクの処理を中央処理部2と同期して並列に実行する中央処理装置(CPU)11と、CPU11の処理結果と中央処理部2のCPU21の処理結果とを比較する比較回路(CMP)12と、CMP12の比較結果によりCPU11の出力のゲート制御を行うゲート回路(GAT)13とを備える確結台な中央処理部である。

【0019】同様に、中央処理部2および予備の中央処理部5はそれぞれBSC20およびBSC50と、CPU21およびCPU51と、CMP22およびCMP52と、GAT23およびGAT53とを備えている。

【0020】なお、現用の中央処理部を構成する中央処理部1および中央処理部2のCMP12およびCMP22はそれぞれ自CPU11およびCPU21の処理結果と他CPU21およびCPU11の処理結果とを比較し、不一致結果を得たときは各GAT13およびGAT23を閉じる制御を行う。

【0021】続いて本実施例の動作について説明する。

【0022】今、1つのタスク処理を実行する場合、中央処理部1と中央処理部2とでこのタスク処理を並列に実行する。

【0023】中央処理部1および中央処理部2では、それぞれBSC10およびBSC20の制御によって2重化バス6を通して図示していないメモリから1つのタスクを受け取る。このとき、中央処理部5は予備の中央処理部となり動作しない。

【0024】CPU11とCPU21とはこの受け取ったタスクの処理を同期して実行し、それぞれ処理結果をCMP12およびCMP22へ送る。

【0025】CMP12では自分のCPU11の処理結果と他のCPU21の処理結果とを比較し、不一致となったときはGAT13を閉じる。同様にCMP22では自分のCPU21と他のCPU11の処理結果とを比較し、不一致となったときはGAT23を閉じる。

【0026】このCMP12またはCMP22で不一致が発生したときは、同様のタスク処理を予備の中央処理部5で再度実行する。

【0027】すなわち、予備の中央処理部5ではBSC50の制御により上記タスクと同じ1つのタスクを受け取ってCPU51で処理を行う。CPU51で実行された処理結果はCMP12およびCMP22に送られ、CMP12およびCMP22はそれぞれCPU11およびCPU21の処理結果とCPU51の処理結果とを再度比較し、一致したときのみそれぞれのGAT13またはGAT23を開けて結果を出力する。

【0028】このように本実施例では、1つのタスク処理に対して3つの中央処理部1、2および5でフォール

(4)

特開2000-155698

5

ト・トレラントを再現している。

【0029】次に、図2は本発明のコンピュータのフォールト・トレラント方式の第2の実施例を示すシステムブロック図で、2つタスク処理を同時に実行するマルチタスク処理の例を示している。

【0030】図2を参照すると、本実施例のコンピュータシステムは、2重化バス6に接続されたそれぞれ疎結合な中央処理部1と中央処理部2とで構成された第1組の現用の中央処理部と、それぞれ疎結合な中央処理部3と中央処理部4とで構成された第2組の現用の中央処理部と、1個の予備の中央処理部5とを備えている。つまり、5つの中央処理部を備えて2つのタスク処理を同時に実行するシステムである。

【0031】各中央処理部、例えば中央処理部1は2重化バス6とインタフェースするバス・コントロール部(BSC)10と、BSC10を通して送られたタスク処理を実行する中央処理装置(CPU)11と、CPU11の処理結果と中央処理部2のCPU21の処理結果とを比較する比較回路(CMP)12と、CMP12の比較結果によりCPU11の出力のゲート制御を行うゲート回路(GAT)13とを備える疎結合な中央処理部である。

【0032】同様に、現用の中央処理部2、3、4および予備の中央処理部5はそれぞれBSC20、30、40および50と、CPU21、31、41および51と、CMP22、32、42および52と、GAT23、33、43および53とを備えている。

【0033】なお、第1組の現用の中央処理部を構成する中央処理部1および中央処理部2のCMP12およびCMP22はそれぞれ自CPU11およびCPU21の処理結果と他CPU21およびCPU11の処理結果とを比較し、不一致結果を得たときは各GAT13およびGAT23を閉じる制御を行う。

【0034】同様に、第2組の現用の中央処理部を構成する中央処理部3および中央処理部4のCMP32およびCMP42はそれぞれ自CPU31およびCPU41の処理結果と他CPU41およびCPU31の処理結果とを比較し、不一致結果を得たときは各GAT33およびGAT43を閉じる制御を行う。

【0035】続いて本実施例の動作について説明する。

【0036】今、タスク処理aとタスク処理bの2つのタスク処理があった場合、タスク処理aを中央処理部1と中央処理部2とで同期して実行し、同時にタスク処理bを中央処理部3と中央処理部4とで同期して実行する。

【0037】中央処理部1と中央処理部2とでは、BSC10とBSC20の制御によって2重化バス6を通して図示していないメモリからそれぞれタスク処理aを受け取る。同様に中央処理部3と中央処理部4とでは、BSC30とBSC40の制御によってそれぞれタスク処

6

理bを上記メモリから受け取る。このとき、中央処理部5は予備の中央処理部となり動作しない。

【0038】CPU11とCPU21とはこの受け取ったタスク処理aを同期して実行し、それぞれ処理結果をCMP12およびCMP22へ送る。

【0039】CMP12では自分の中央処理部1内のCPU11の処理結果と同じ組の他の中央処理部2内のCPU21の処理結果とを比較し、不一致となったときはGAT13を閉じる。同様にCMP22では自分の中央処理部2内のCPU21の処理結果と同じ組の他のCPU11の処理結果とを比較し、不一致となったときはGAT23を閉じる。

【0040】タスク処理bも同様に処理が実行される。すなわち、CPU31とCPU41とは受け取ったタスク処理bを同期して実行し、それぞれ処理結果をCMP32およびCMP42へ送る。

【0041】CMP32では自分のCPU31の処理結果と同じ組の他のCPU41の処理結果とを比較し、不一致となったときはGAT33を閉じる。同様にCMP42では自分のCPU41の処理結果と同じ組の他のCPU31の処理結果とを比較し、不一致となったときはGAT43を閉じる。

【0042】今、中央処理部3と中央処理部4とでタスク処理bを実行したときに不一致が発生したとすれば、タスク処理bを予備の中央処理部5で再度実行する。

【0043】すなわち、予備の中央処理部5ではBSC50の制御により受け取ったタスク処理bをCPU51で実行する。CPU51で実行された処理結果はCMP32とCMP42とに送られ、CMP32およびCMP42はそれぞれCPU31の処理結果およびCPU41の処理結果とCPU51の処理結果とを再度比較し、一致したときのみそれぞれのGAT33またはGAT43を開けて結果を出力する。

【0044】また、タスク処理aで処理結果に不一致が発生したときも、同様に予備の中央処理部5でタスク処理aを再度実行する。

【0045】このように第2の実施例では、2つのタスク処理に対して5つの中央処理部1、2、3、4および5でフォールト・トレラントを再現している。

【0046】なお、同時に処理するタスク処理数がn個(3個以上)のときにも、 $2n+1$ 個の中央処理部により同様の動作でマルチタスク処理のフォールト・トレラントを再現する。

【0047】また、第1、第2の実施例において、予備の中央処理部5は現用の中央処理部1、…、4と同じ構成とするためCMP52およびGAT53を備えているが、中央処理部5が上記のタスク処理を再度実行したときCMP52およびGAT53は動作せず、CPU51の処理結果のGAT53からの出力は行われない。

【0048】

(5)

特開2000-155698

7

8

【発明の効果】以上説明したように本発明は、疎結合な中央処理部を使用し、通常は1つのタスク処理に対して2つの中央処理部を使用して同期して処理し、この中央処理部の一方に障害が発生したときのみ、予備の中央処理部を使用して処理を継続することにより、マルチタスク処理の場合にも $(2 \times \text{タスク処理数} + 1)$ 個の中央処理部数でフォールト・トレラント動作を実現することができるので、コンピュータシステムのハードウェア構成を物理的に小規模にすることができるという効果を有する。

【図面の簡単な説明】

【図1】本発明のコンピュータのフォールト・トレラント方式の第1の実施例を示すシステムブロック図である。

【図2】本発明のコンピュータのフォールト・トレラント方式の第2の実施例を示すシステムブロック図である。

*【図3】従来のコンピュータのフォールト・トレラント方式の第1の例を示すシステムブロック図である。

【図4】従来のコンピュータのフォールト・トレラント方式の第2の例を示すシステムブロック図である。

【符号の説明】

1, 2, 3, 4, 5, 100 中央処理部

6 2重化バス

10, 20, 30, 40, 50 バス・コントロール部(BSC)

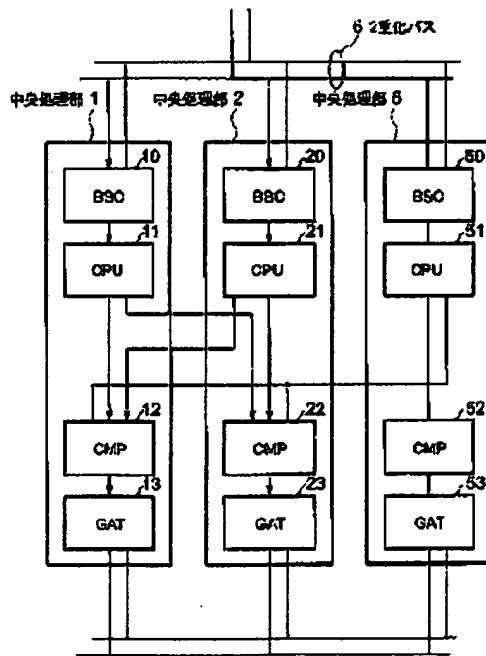
10 11, 21, 31, 41, 51, 101, 102, 103, 411, 412, 413 中央処理装置(CPU)

12, 22, 32, 42, 52, 104, 414, 415, 416 比較回路(CMP)

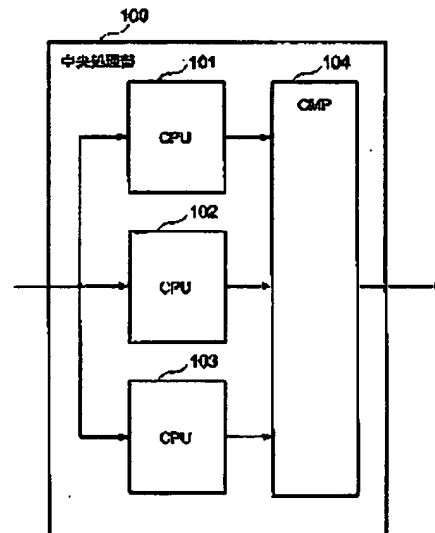
13, 23, 33, 43, 53, 418, 419, 420 ゲート回路(GAT)

* 417 出力バス

【図1】



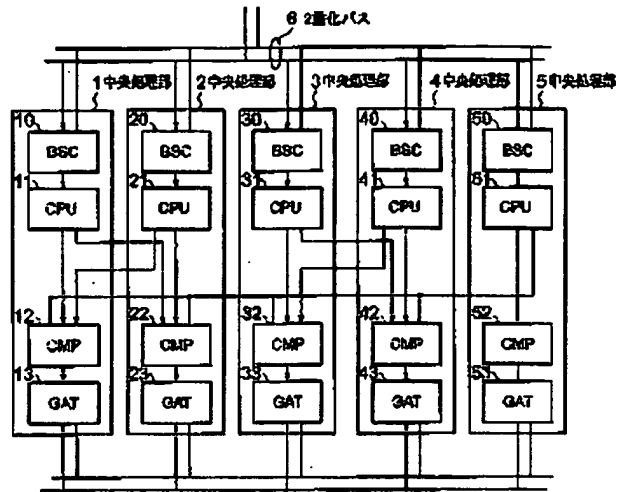
【図3】



(6)

特開2000-155698

【図2】



【図4】

